Yandex

Speeding up prediction performance of the BDT-based models.

Egor Khairullin, Andrey Ustyuzhanin

Introduction

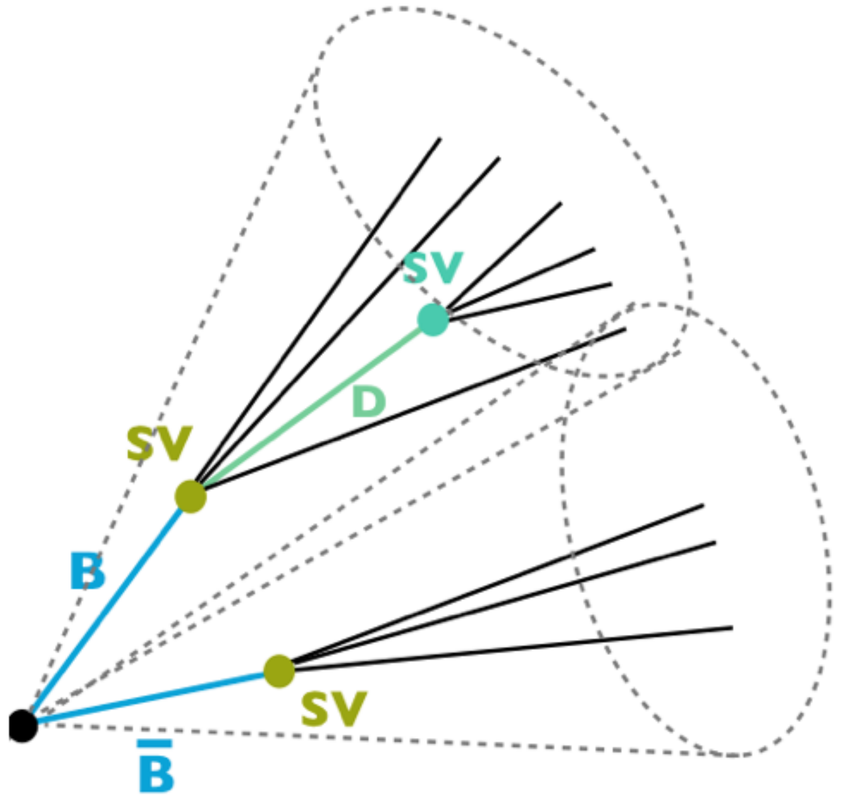
Online Trigger:

〉 Sample: one proton-proton collision

〉 Binary classification: is event interesting or not (B decay)

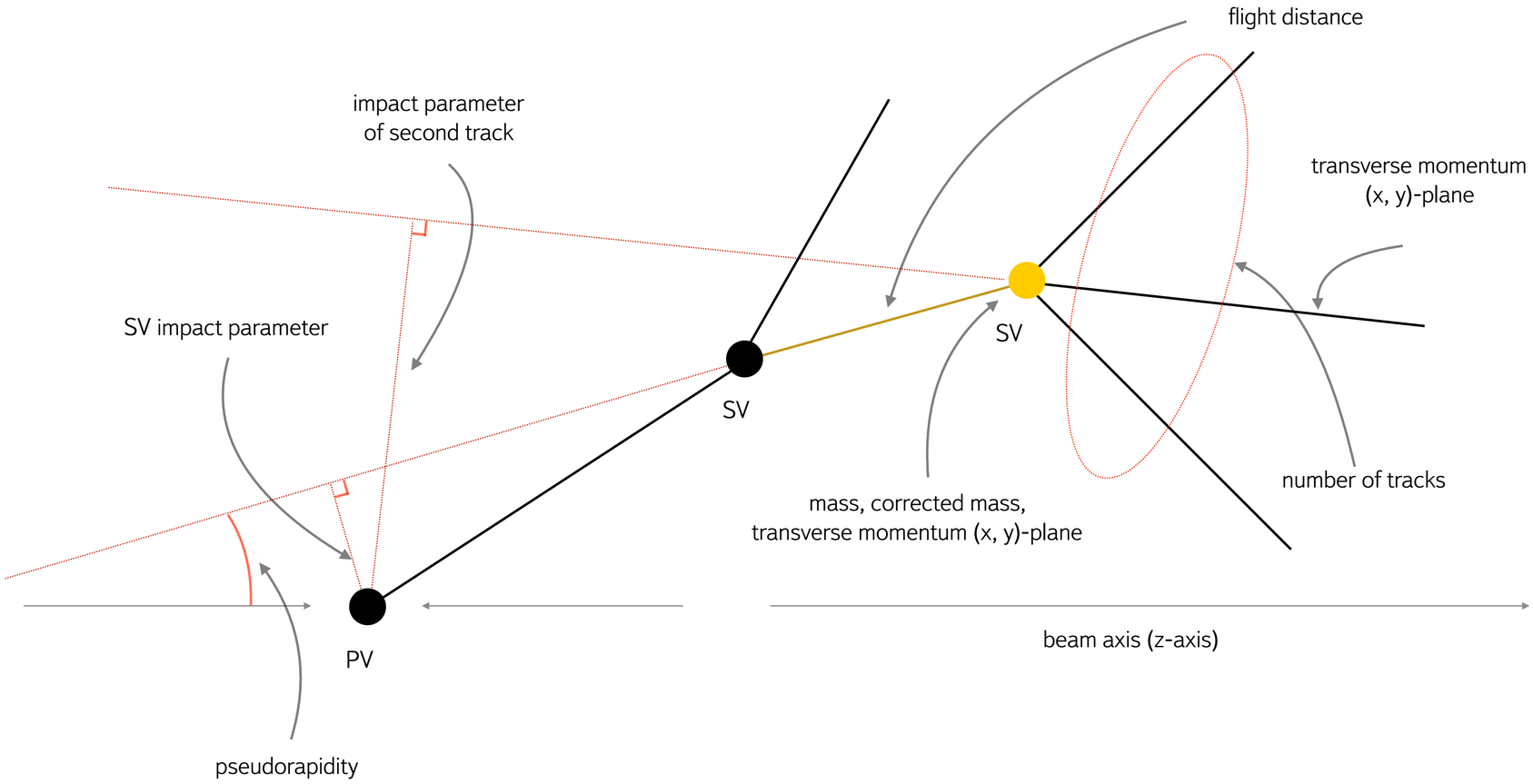
〉 Event consists of:

1. secondary vertices (SV)
2. tracks (track description)



http://arxiv.org/abs/1510.00572

Data features



Machine learning problem

"Signal":

simulated events for decays channels of interest

"Background":

simulated generic proton-proton collisions

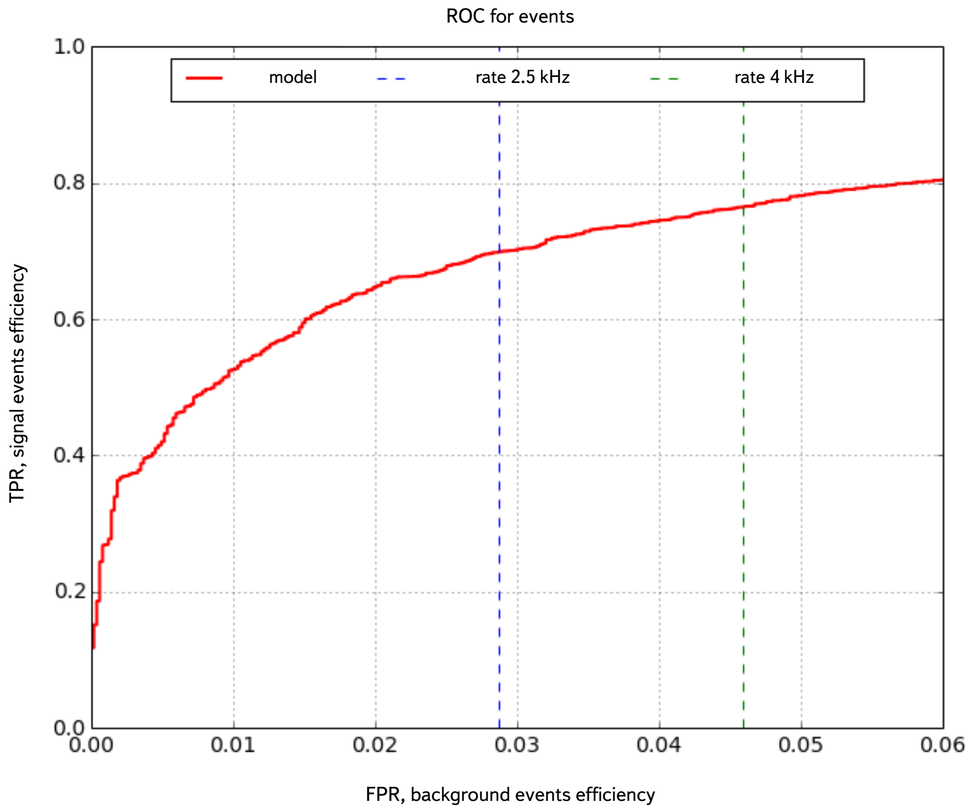
Goal:

get the highest signal efficiency given background rejection rate

Figure of Merit

〉 Optimize Area under ROC curve in a region with small False Positive Rate (<0.05)

〉 Corresponds to mean efficiency of the model under given bandwidth limit



Restrictions

- Real-time event processing (time restriction, apply model to events one by one)
- Limited memory

BDT & DTensor

BBDT (DecisionTensor)

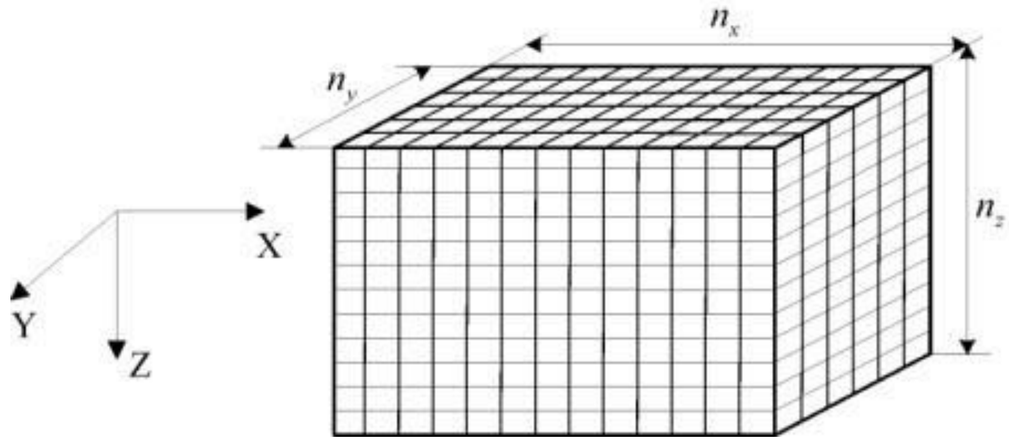
〉 Convert decision trees to ***n***-dimensional table (lookup table), ***n*** is number of features

〉 Very fast

〉 Number of bins for each feature should be small (otherwise size grows too big)

〉 https://arxiv.org/abs/1210.6861

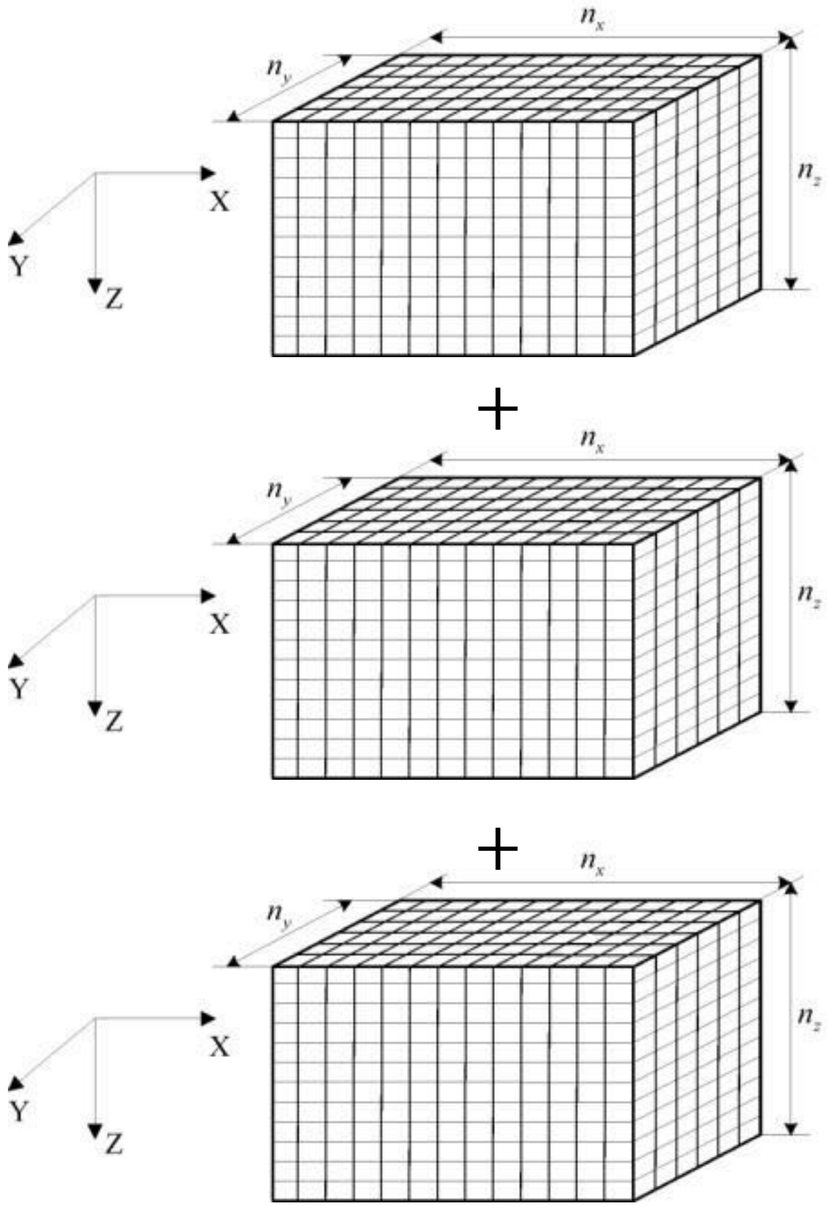
〉 $$D = (f_1, ..., f_n)$$
$$f_i = \{b_{i1}, ..., b_{ip_i}\}$$
$$S(D) \propto \prod |f_i| = \prod p_i$$



\\
D = (f_{1}, ..., f_{n}) \\
f_{i} = \{b_{i1}, ..., b_{ip_{i}}\} \\
S(D) \propto \prod |f_{i}| = \prod p_i

DecisionTensor Ensemble

- Several DecisionTensors with different bins
- Slower than one DecisionTensor
- Higher quality could be achieved
- **How to choose ensemble parameters?**



DecisionTensor Similarity

$$D^1 = (f_1^1, \dots, f_n^1)$$
$$D^2 = (f_1^2, \dots, f_n^2)$$

$$D^u = D^1 + D^2$$
$$D^u = (f_1^u, \dots, f_f^u), f_i^u = f_i^1 \cup f_i^2$$

$$Sim(D^1, D^2) = S(D^1) + S(D^2) - S(D^1 + D^2)$$

Merging trees into DecisionTensor Ensemble

Given N, K:

1. Build BDT that consists of N trees
2. Initialize K empty DecisionTensors.
3. Take next tree T from BDT model, make new DecisionTensor from it and merge it with the most similar DecisionTensor.
4. Repeat 3. until all trees has been used.

CatBoost

Yandex
CatBoost

- CatBoost is an open-source gradient boosting library (BDT) from Yandex
- Uses oblivious trees
- Discretize features
- Can improve existing model (baseline)
- Trained model usually consists of hundreds trees (and may be slow)
- https://github.com/catboost

CatBoost to DecisionTensor Ensemble (DTE)

1. Choose number of DTensors (K) by time restrictions;
2. Train CatBoost with few bins (memory restrictions) for each feature. Use it as Baseline for CatBoost models generated later;
3. Merge all trees into first DecisionTensor (D_1) of Ensemble;
4. Random search parameters for CatBoost;

   Train CatBoost model
   Convert model to DTE of size K-1
   Remove DTEs exceeding memory restrictions

5. Choose DTE with the best quality on the test sample.

Experiment

Setup

〉 Data

Mean to recompute event prediction from SV predictions
1M SVs, 160K events, 10 features
Train - 50%, Test - 25%, Validation Set - 25%

〉 Up to 1 GB Memory

〉 Measure time of inference, applying model to each SV (one by one)
of the whole Validation Set.

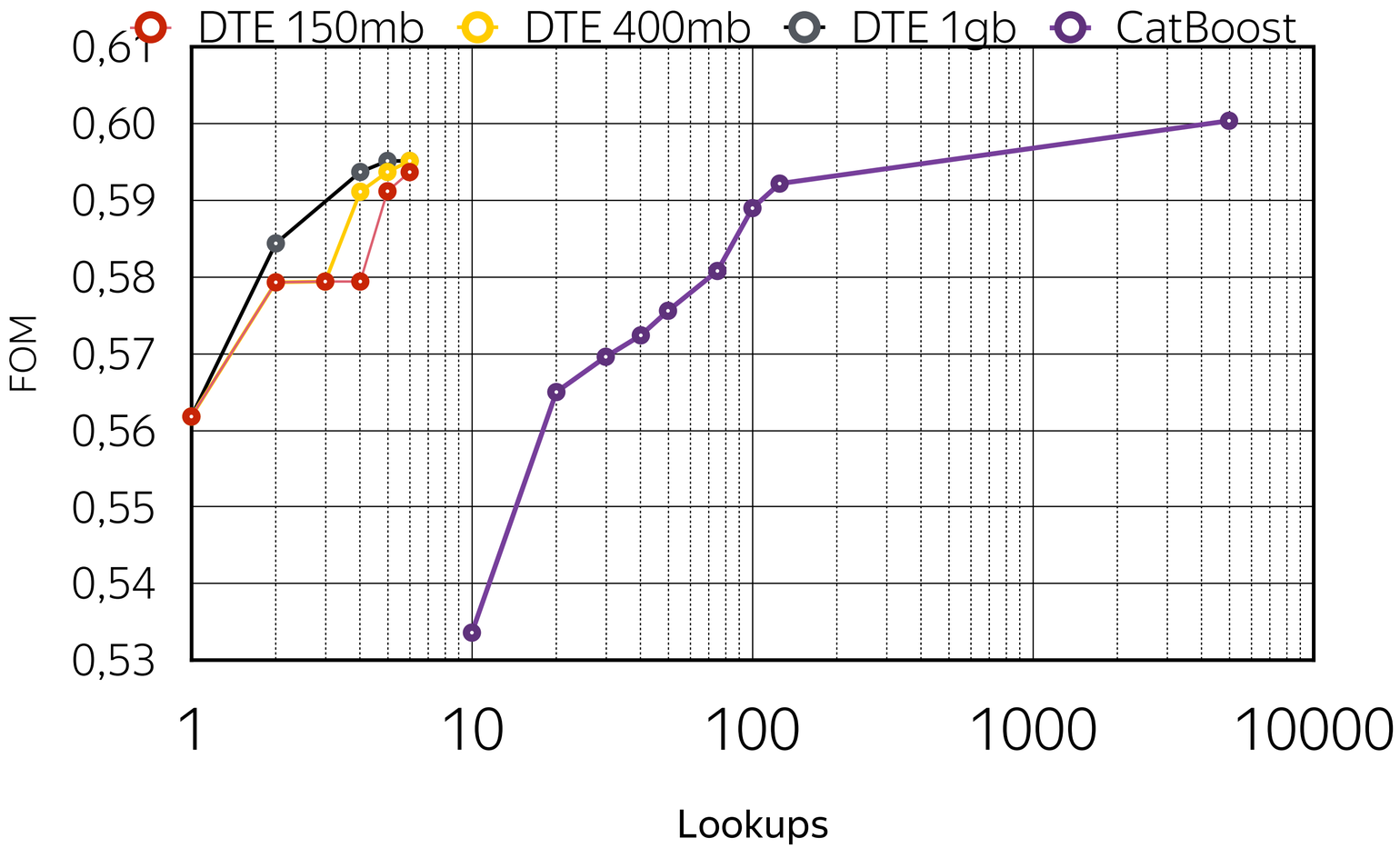
Lookups vs Time

CatBoost Best: 5000 trees

CatBoost Fast: 10-125 trees

DTE: 1-6 DTs

DTE size 150mb-1gb

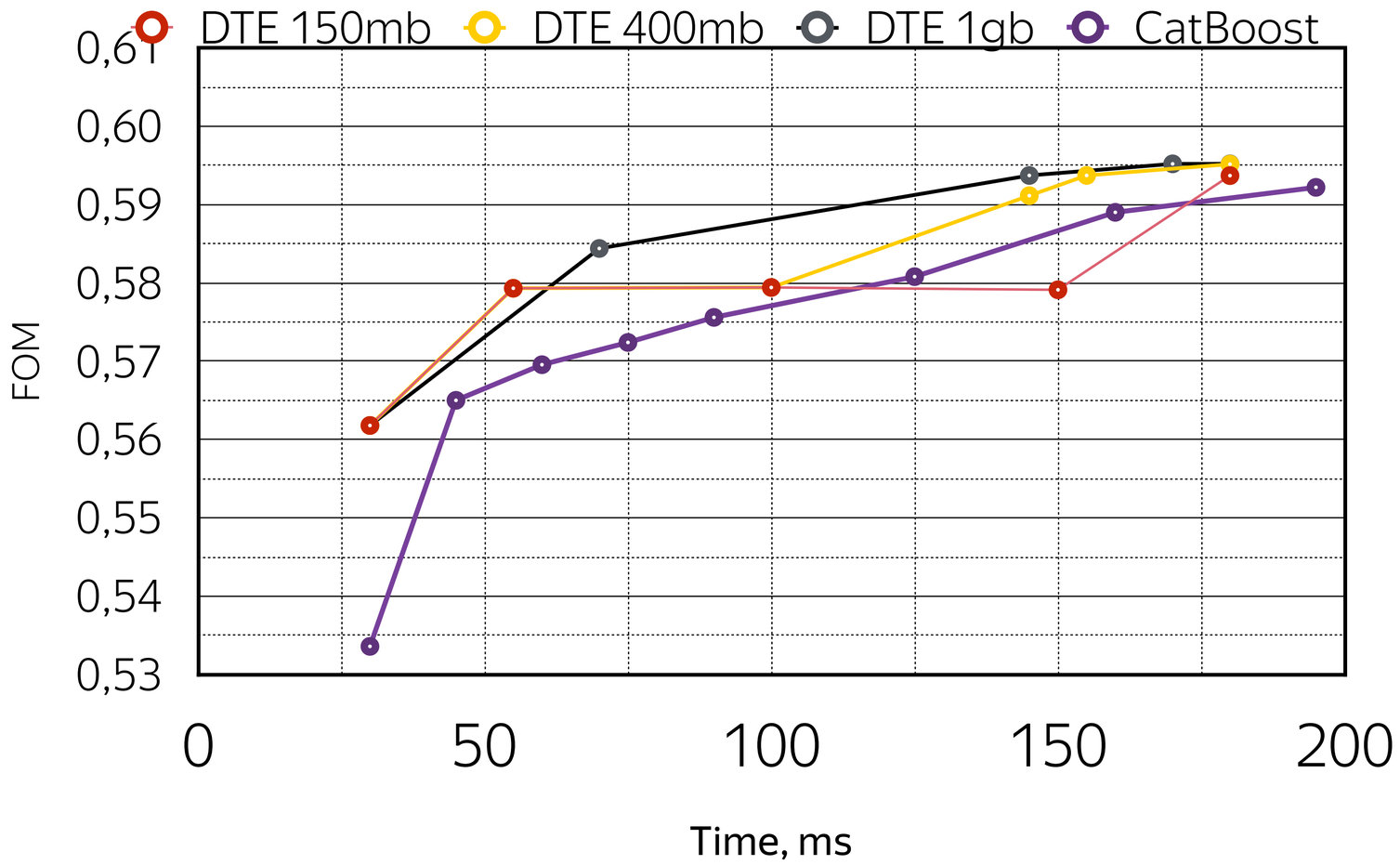


Quality vs Time

CatBoost Fast: 10-125 trees

DTE: 1-6 DTs

DTE size 150mb-1gb



DTE vs CatBoost

〉 Equal FOM:

DTE is up to 2,5 times faster and has up to 25 times fewer lookups

〉 "Best" CatBoost model (5000 trees):

DTE is 50 times faster and has 1000 less lookups
Difference in FOM only 0.006 (less than 1 per mil)

〉 Memory access is slow so DTensor lookups cost more than simple tree lookups (which can be cached)

〉 Faster memory (DDRAM5? FPGA?) could speed up DTensor lookups

Summary

DecisionTensor Ensemble

- helps running complex BDT (such as CatBoost) in real-time
- improves quality compared to single BBDT
- gives trade-off memory vs speed with marginal quality decrease

Thanks for attention!

Egor Khairullin
Lead software engineer

 mikari@yandex-team.ru

Andrey Ustyuzhanin
Head of research group

 anaderi@yandex-team.ru

Comparison

Model	ROC	Time, ms	Size, Mb
catboost - 5000	0,6004	10000	<5
catboost - 100	0,589	160	<5
catboost - 75	0,5808	125	<5
catboost - 10	0,5336	30	<5
big dte - 6	0,5952	180	1000
big dte -3	0,5938	145	1000
big dte -2	0,5844	70	1000
small dte - 2	0,5794	55	150
dte - 1	0,5618	30	150

Decays

mode	2.5 kHz	4. kHz
$B^0 \to K^*[K^+\pi^-]\mu^+\mu^-$	1.64	1.72
$B^+ \to \pi^+K^-K^+$	1.59	1.65
$B_s^0 \to D_s^-[K^+K^-\pi^-]\mu^+\nu_\mu$	1.14	1.47
$B_s^0 \to \psi(1S)[\mu^+\mu^-]K^+K^-\pi^+\pi^-$	1.62	1.71
$B_s^0 \to D_s^-[K^+K^-\pi^-]\pi^+$	1.46	1.52
$B^0 \to D^+[K^-\pi^+\pi^+]D^-[K^+\pi^-\pi^-]$	1.40	1.86

CatBoost to DecisionTensor Ensemble

- Choose count of DTensors by time restrictions.
- First DecisionTensor:

  Train CatBoost with few bins (memory restrictions) for each feature
  Merge all trees into first DecisionTensor (D_1) of Ensemble

- Next DecisionTensors:

  Train CatBoost with default bins but low tree count (due to memory and time restrictions)
  Merge trees to DecisionTensors ($D_2, D_3, ..., D_k$).

Models to compare

- CatBoost default model, 5000 trees
- CatBoost small models: 10-125 trees
- DT Ensemble: size from 1 to 6